

日 本 国 特 許 庁  
JAPAN PATENT OFFICE

別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office.

出 願 年 月 日                      2 0 0 3 年    8 月 2 8 日  
Date of Application:

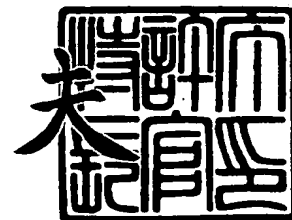
出 願 番 号                      特 願 2 0 0 3 - 3 0 5 2 5 8  
Application Number:  
[ST. 10/C] :                      [ J P 2 0 0 3 - 3 0 5 2 5 8 ]

出      願      人                      松 下 電 器 産 業 株 式 会 社  
Applicant(s):

2 0 0 3 年    9 月 1 6 日

特許庁長官  
Commissioner,  
Japan Patent Office

今 井 康



出証番号    出証特 2 0 0 3 - 3 0 7 5 9 2 4

【書類名】 特許願  
【整理番号】 2030750072  
【提出日】 平成15年 8月28日  
【あて先】 特許庁長官殿  
【国際特許分類】 G06F 12/00  
【発明者】  
    【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内  
    【氏名】 高木 利匡  
【発明者】  
    【住所又は居所】 大阪府門真市大字門真 1 0 0 6 番地 松下電器産業株式会社内  
    【氏名】 前田 茂則  
【発明者】  
    【住所又は居所】 ドイツ国 ランゲン 6 3 2 2 5 モンツァシュトラーク 4 シー  
    ー パナソニック ヨーロピアン ラボラトリーズ ゲーエムベ  
    ーハー内  
    【氏名】 ヨーク フォグラール  
【発明者】  
    【住所又は居所】 ドイツ国 ランゲン 6 3 2 2 5 モンツァシュトラーク 4 シー  
    ー パナソニック ヨーロピアン ラボラトリーズ ゲーエムベ  
    ーハー内  
    【氏名】 ゲラルド ファイファー  
【特許出願人】  
    【識別番号】 000005821  
    【氏名又は名称】 松下電器産業株式会社  
【代理人】  
    【識別番号】 100105050  
    【弁理士】  
    【氏名又は名称】 鷲田 公一  
【先の出願に基づく優先権主張】  
    【出願番号】 特願2002-284164  
    【出願日】 平成14年 9月27日  
【手数料の表示】  
    【予納台帳番号】 041243  
    【納付金額】 21,000円  
【提出物件の目録】  
    【物件名】 特許請求の範囲 1  
    【物件名】 明細書 1  
    【物件名】 図面 1  
    【物件名】 要約書 1  
    【包括委任状番号】 9700376

**【書類名】 特許請求の範囲****【請求項 1】**

1 以上のコンポーネントを保持するデータベースを有しており、前記コンポーネントを読み込んだファイルに適用して処理を施すデータ処理装置であって、前記データベース内には前記ファイルのファイル情報に基づいて処理に適したコンポーネントを判定するタイプ判定関数を有することを特徴とするデータ処理装置。

**【請求項 2】**

前記タイプ判定関数は、前記コンポーネント各々の内部に保持されていることを特徴とする請求項 1 記載のデータ処理装置。

**【請求項 3】**

前記データベースは、前記タイプ判定関数の参照先を示す判定関数エントリを保持することを特徴とする請求項 1 または請求項 2 記載のデータ処理装置。

**【請求項 4】**

前記判定関数エントリは、各コンポーネントと 1 対 1 に対応して保持されることを特徴とする請求項 3 記載のデータ処理装置。

**【請求項 5】**

前記データベースに保持されるコンポーネントの中から処理に適したコンポーネントを検索するための判定手段をさらに有し、

前記判定手段は、前記判定関数エントリを前記データベースから読み込み、さらに前記判定関数エントリが参照先として示す前記タイプ判定関数を前記データベースより読み込み、処理対象であるファイルのファイル情報を前記判定関数に適用して判定することより、処理に適したコンポーネントを検索することを特徴とする請求項 3 または請求項 4 記載のデータ処理装置。

**【請求項 6】**

前記判定手段により検索されたコンポーネントを読み込んで組合せることにより、目的とする処理を実行する処理部をさらに備え、

前記処理部内に存する制御手段は、コンポーネントの出力ファイルのファイル情報を読み込んで前記判定手段に伝えることにより、前記出力ファイルに対する次の処理に適したコンポーネントを選択することを可能とすることを特徴とする請求項 5 記載のデータ処理装置。

**【請求項 7】**

過去に構成したコンポーネントの構成パターンを記憶するコンポーネント構成データベースと、読み込んだファイルの情報に基づいて、記憶された構成パターンの中から処理に適した構成パターンを検索する構成パターン検索手段とを有することを特徴とする請求項 1 記載のデータ処理装置。

**【請求項 8】**

コンポーネントを格納したデータベースサーバにネットワークを介して接続可能なダウンロード手段を備え、検索された構成パターンに含まれるコンポーネントが前記データベースに存在しない場合は、前記ダウンロード手段が前記データベースサーバから必要なコンポーネントを取得することを特徴とする請求項 7 記載のデータ処理装置。

**【請求項 9】**

前記ダウンロード手段は、データ処理装置に接続されたコンポーネントを格納した記録媒体から必要なコンポーネントを取得することを特徴とする請求項 8 記載のデータ処理装置。

**【請求項 10】**

前記ダウンロード手段が新たなコンポーネントを取得したとき、取得したコンポーネントは前記データベースに格納され、当該コンポーネントの格納先を示す情報は前記コンポーネント構成データベースに保持されることを特徴とする請求項 8 記載のデータ処理装置。

**【請求項 11】**

対象とするデータの処理に適したコンポーネントであるか否かを判定するタイプ判定関数と前記タイプ判定関数を参照先として出力する判定関数エントリとを保持するデータベースを備えたデータ処理装置のデータ処理方法であって、

前記判定関数エントリを前記データベースから読み込む第 1 のステップと、

前記判定関数エントリが参照先として示す前記タイプ判定関数を前記データベースより読み込む第 2 のステップと、

前記タイプ判定関数に処理対象であるデータの情報を適用し、適用可否の結果を得る第 3 のステップと、

を有し、処理に適したコンポーネントが選択されるまで前記第 1 のステップから第 3 のステップを繰り返すことを特徴とするデータ処理方法。

【請求項 12】

対象とするデータの処理に適したコンポーネントをデータベースから選択するためのデータ処理プログラムであって、コンピュータに、

対象とするデータの処理に適しているコンポーネントか否かを判定するタイプ判定関数の参照先を出力する判定関数エントリを読み込むステップと、

前記判定関数エントリが参照先として示すタイプ判定関数を読み込むステップと、

前記タイプ判定関数に処理対象であるデータの情報を適用し、適用可否の結果を得るステップと、

を実行させるためのデータ処理プログラム。

【書類名】明細書

【発明の名称】データ処理装置、データ処理方法及びデータ処理プログラム

【技術分野】

【0001】

本発明は、ある機能を実現するために組み合わせられたソフトウェア部品から構成されるコンポーネントを読み込んで処理を実行するデータ処理装置、およびコンポーネントデータベースに関し、特にメディアデータ処理コンポーネントの保持方法および高速な読み出し方法に関する。

【背景技術】

【0002】

従来のデータ処理装置は、データベースにコンポーネントの名前および属性が保持されており、データ処理制御部は目的のコンポーネントを名前や属性より探し出し、また名前や属性の情報では不足している場合、実際にコンポーネント全体を読み出すように構成されているものがある（例えば、特許文献1の図1参照）。

【0003】

特許文献1に記載されているデータ処理装置（文献中では、資源管理装置と呼ばれている）の全体構成を図20に示す。

【0004】

同図に示すデータ処理装置800は、コンテキスト保持部801-1～N、属性保持部802-1～N、名前解析部803-1～N、名前変換部804-1～N、名前解決部805、資源実現部806、資源保持部807-1～M、資源管理部808、通信部809、コンテキストデータベース部810、コンテキストデータベース検索部811、コンテキスト選択部812、及びコンテキスト抽出部813から構成されている。

【0005】

コンテキスト保持部801-1～801-Nには、属性保持部802-1～802-N、名前解析部803-1～803-N、名前変換部804-1～804-Nがそれぞれ設けられている。属性保持部802-1～802-Nには、それぞれコンテキスト保持部801-1～801-Nに付けられた任意の個数の属性が、その属性名と属性値のペアのリストとして保持されている。

【0006】

名前解析部803-1～803-Nは、それぞれ入力される資源の名前を解析する。名前変換部804-1～804-Nは、それぞれ名前解析部803-1～803-Nで解析された結果を受け、解析結果を資源実現表現に変換する。この資源実現表現は、資源検索式を要素とする資源要素列と、その資源要素列を処理するための手続き情報との組から構成される。ここで、資源検索式は、計算機システム内にあらかじめ実現されている実資源を指定するための検索式である。

【0007】

名前解決部805は、コンテキスト保持部801-1～801-Nが出力した資源実現表現を受け取り、その資源実現表現のすべてあるいはいづれかに対応する名前解決結果が資源実現部806に対して出力される。

【0008】

資源実現部806は、名前解決部805から出力された名前解決結果を入力し、その名前解決結果に対応する資源のハンドルを出力する。

【0009】

資源保持部807-1～Mは、それぞれあらかじめ実現されている資源を保持するとともに、その資源を識別するための識別子を保持している。

【0010】

資源管理部808は、資源保持部807-1～Mが保持している資源を管理するとともに、前記資源検索式に従って資源を特定する。

【0011】

通信部 809 は、名前解決部 805、資源実現部 806、資源管理部 808、及びコンテキストデータベース部 810 間の通信を行う。

【0012】

コンテキストデータベース部 810 は、N 個のコンテキスト保持部 801-1~N をコンテキスト集合として管理しており、コンテキストデータベース検索部 811、コンテキスト選択部 812、コンテキスト抽出部 813 を有している。

【0013】

コンテキストデータベース検索部 811 は、属性を用いて記述されているコンテキスト検索式を受け取り、そのコンテキスト検索式を満足する属性を有しているコンテキスト保持部 801-i (i は 1~N の整数) をコンテキスト集合から探し出す。検索結果は、例えばコンテキスト保持部 801-i に対するリファレンスの集合として出力することができる。

【0014】

コンテキスト選択部 812 は、コンテキストデータベース間等の通信を行う通信部 809 にコンテキスト集合の抽出を行うためのコンテキスト検索式が入力されたとき、このコンテキスト検索式をコンテキストデータベース検索部 811 に渡して、コンテキスト検索式を満足するコンテキスト保持部 801-i の集合を得て、コンテキスト抽出部 813 に出力する。

【0015】

コンテキスト抽出部 813 は、コンテキスト保持部 801-i に対するリファレンスの集合を入力とし、そのリファレンスによって示されるコンテキスト保持部 801-i と等価なコンテキスト保持部を合成するために必要な情報をコンテキスト表現として出力する。

【0016】

例えば、データ処理装置 800 において、属性保持部 802-1~N には、所有者を表わす属性 “owner”、変更日時を表わす “mdate”、及びカテゴリを表わす “category” の 3 つの属性が、属性名と属性値のペアのリストとして保持されているものとする。ここでは、属性保持部 802-1 の “owner” の値は “Smith”、“mdate” の値は 95. 3. 18、及び “category” の値は “picture” であり、属性保持部 802-2 の “owner” の値は “Smith”、“mdate” の値は 95. 11. 14、及び “category” の値は “picture” であり、属性保持部 802-3~N の全ての “owner” の値は “Bill”、“mdate” の値は 95. 3. 3、及び “category” の値は “sound” であるとして、コンテキストの抽出を行う動作を中心に説明する。

【0017】

まず、例えば所有者を表わす属性 “owner” の値が “Smith” に等しく、かつ、変更日時を表わす属性 “mdate” の値が 95. 3. 16 よりも大きく、かつ、カテゴリ属性 “category” の値が “picture” に等しいコンテキスト保持部 801 を検索する場合を考える。

【0018】

コンテキスト選択部 812 は、コンテキストデータベース検索部 811 に上記検索を実現するための検索式を渡し、この検索式を満足する属性を属性保持部 802 に有するコンテキスト保持部 801 を検索する。検索式は例えば、C++ Object Query Language に準拠した場合は図 21 に示すような記載となる。この検索式をコンテキスト保持部 801-1 およびコンテキスト保持部 801-2 のみが満たすので、検索結果としてコンテキストデータベース検索部 811 は、コンテキスト保持部 801-1 のリファレンスと、コンテキスト保持部 801-2 のリファレンスとからなる集合をコンテキスト選択部 812 に返す。

【0019】

コンテキスト保持部 801-1 のリファレンスと、コンテキスト保持部 801-2 のリ

ファレンスとからなる集合を受け取ったコンテキスト選択部 812 は、受け取った集合をコンテキスト抽出部 813 に入力し、リファレンスに該当するコンテキスト保持部 801-1 及びコンテキスト保持部 801-2 を取り出すよう要求することにより、該当するコンテキストを取得することができる。

【特許文献 1】特開平 9-171501 号公報

【発明の開示】

【発明が解決しようとする課題】

【0020】

しかしながら、従来のデータ処理装置では、従来の名前や属性では識別できないような新規の属性が付加されたコンテキストが追加された場合、それに合わせて従前のコンテキスト保持部も再構築しなければならないという問題があった。

【0021】

具体的には、例えば、ファイルサイズを示す属性“size”が追加された場合は、全てのコンテキスト保持部を更新する必要がある、コンテキスト保持部の数が多い場合や、頻繁に新しい属性が追加されている場合は、コンテキスト保持部のメンテナンスの負荷が大きくなってしまいう問題が発生する。

【0022】

本発明は、このような問題を解決するためになされたもので、新規コンポーネントが追加された場合でもデータベースの再構築を不必要とするデータ処理装置、データ処理方法及びデータ処理プログラムを提供することを目的とする。

【課題を解決するための手段】

【0023】

上記課題を解決するために本発明においては、コンポーネント内にコンポーネント本体とは別に、対象とするデータの処理に適するコンポーネントであるか否かを判定するタイプ判定関数を有し、データ処理装置には、読み込んだファイルまたは前段のコンポーネントにより処理され出力されたデータの情報を前記タイプ判定関数に適用する判定手段を備え、各コンポーネントと対応するタイプ判定を実施することで、判定のための属性が変更または追加となったとしても、タイプ判定関数のみを変更すれば足り、コンポーネント本体（従来のコンテキスト保持部全体に相当）を更新する必要はないため、効率的にデータベースを更新することが可能となる。

【発明の効果】

【0024】

以上説明したように、本発明によれば、新規コンポーネントが追加された場合でもデータベースの再構築を不必要とすることができる。

【発明を実施するための最良の形態】

【0025】

以下、本発明の実施の形態について図 1 から図 19 を用いて説明する。なお、本発明は本実施の形態に何ら限定されるものではなく、その要旨を逸脱しない範囲において、様々な態様で実施し得る。

【0026】

（実施の形態 1）

図 1 は、本発明の実施の形態 1 におけるデータ処理装置 100 を示したブロック図である。図 1 において、メディアコンポーネントデータベース 101 は、メディアを処理するためのコンポーネントであるメディアコンポーネントを保持している。メディア判定部 102 は、メディアコンポーネントを比較して判別する。操作入力部 103 は、ユーザからの入力を受け付ける。ファイルシステム 104 は、マルチメディアデータを保持している。メディア処理部 105 は、メディアの処理を行う。映像表示部 111 は、映像を表示する。音声出力部 112 は、音声を出力する。

【0027】

図 2 は、メディアコンポーネントデータベース 101 が保持するコンポーネントの保持

形態を示した図である。

【0028】

メディアコンポーネントデータベース101には、メディアコンポーネント30-1～Nが保持されており、各メディアコンポーネントの中にはそのコンポーネントがどのような種類のデータに適用されるべきコンポーネントであるかを判定するためのメディアタイプ判定関数40-1～Nが含まれている。また、メディアコンポーネントデータベース101には、メディアタイプ判定関数40-1～Nの各々への参照先アドレスを返す関数であるメディアコンポーネントメディアタイプ判定関数エントリ20-1～Nが保持されている。

【0029】

以上のように構成されたデータ処理装置100について、図3、図4を用いてその動作を説明する。

【0030】

図3は、データ処理装置100がマルチメディアファイルを再生する手順を示したフロー図である。

【0031】

データ処理装置100のユーザからファイルの再生要求が操作入力部103に入力され、ファイル名と再生要求がメディア処理部105内のメディア制御部106に送られる（S1010）。

【0032】

メディア制御部106は、指定されたファイルを読み込むためのコンポーネント（ファイル読込コンポーネント）をメディア判定部102へ要求する（S1020）。

【0033】

メディア判定部102は、指定されたファイル名をもとにファイルを読み込むためのコンポーネント（ファイル読込コンポーネント）をメディアコンポーネントデータベース101より検索する（S1030）。

【0034】

この検索を行う際は、図4に示すフロー図に従った処理を行う。

【0035】

すなわち、まず、メディア判定部102は、メディアコンポーネントデータベース101よりメディアタイプ判定関数エントリ20-1を読み込む（S2010）。なお、図4において、xの初期値は1であるものとする。次に、読み出したメディアタイプ判定関数エントリ20-1からの参照先であるメディアコンポーネント30-1のメディアタイプ判定関数40-1と、ユーザが操作入力部103を介して指定したファイル名に対するファイル名とを用いて、このメディアコンポーネント30-1が要求されたファイルの読み込みに適しているか否かが判定される（S2020）。

【0036】

メディアタイプ判定関数40-1の判定の結果、メディアコンポーネント30-1が、ユーザが指定したファイル名に対応するデータの処理コンポーネントとして適していなかった場合は、xの値を1だけ増加させ（S2040）、次のメディアタイプ判定関数エントリ20-2を読み込む（S2010）。一方、処理コンポーネントとして適していた場合は、メディア判定部102は、メディアコンポーネントデータベース101よりメディアコンポーネント30-1を読み出し、メディア制御部106へ返す（S2030）。

【0037】

メディア判定部102にてこのような処理を行うことによって、メディアコンポーネント全体を読み出すことなく、ユーザ指定のデータの処理コンポーネントとして適しているか否かの判定を行うことができ、メディアコンポーネントの検索の高速化を図ることができる。

【0038】

また、メディア判定のための方法が追加や変更された場合でも、コンポーネント内のメ



メディアタイプ判定関数 40-1~N のみの変更で済み、データベース自体の再構築が不要となる。すなわち、例えば、新しいタイプのメディアデータを処理するコンポーネントが追加された場合、従来の技術ではコンテキスト保持部を再構築することが必要となるが、本願発明のデータ処理装置は、各コンポーネント本体は再構築する必要がなく、メディアタイプ判定関数 40-1~N のみを変更することによって、新しいタイプのメディアデータの処理が可能であるか否かを判別することが可能となる。

#### 【0039】

図3のフロー図を参照して、メディア制御部106は、読み込んだファイル読込コンポーネント107を用い、ファイルシステム104よりファイルを読み込む(S1040)。

#### 【0040】

メディア制御部106は、読み込んだファイルの情報から多重化情報を取得し、その逆多重化を行うためのコンポーネント(逆多重化コンポーネント108)をメディア判定部102へ要求する(S1050)。

#### 【0041】

なお、多重化情報とは、ファイルシステム104に保持されているファイルに付加されている情報であって、そのファイルに多重化されている複数のデータタイプが記載されている情報である。以下、多重化情報について具体的に説明する。

#### 【0042】

図5は、多重化されたファイルの状態を概念的に示した図である。多重化ファイル(MP4 File)には、大きく分けてファイルの書誌的な情報を記載している領域(Movie Atom)と、メディアの実体を格納している領域(Media Atom)とに分かれる。

#### 【0043】

なお、“Movie Atom”領域の構造は、いわゆる入れ子構造となっており、“Movie Atom”領域の中は、さらに“Movie Header Atom”と“Track Atom”といった複数の領域に区別されて情報が保持されている。

#### 【0044】

メディア制御部106は、上記“Movie Atom”の中からメディアタイプ判定関数40-1~Nの引数として必要な書誌的情報を、上記多重化ファイルの記載箇所を特定した情報として取得する。この情報を多重化情報と呼ぶ。

#### 【0045】

図6は、メディア制御部106が取得する多重化情報の例を示したものである。図6において、例えば「メディアタイプ」は、多重化ファイル中の“Movie Atom”の中の“Track Atom”の中の“Media Atom”の中の“Handler Reference”に記載されていることを示している。

#### 【0046】

なお、ここでは指定されたファイルが、メディアデータを多重化しているマルチメディアデータであることにするが、本発明は、このデータに限定されるものではない。

#### 【0047】

再度、図3のフロー図を参照して、メディア判定部102は、メディア制御部106から入力された多重化情報をもとに、逆多重化を行うためのコンポーネント(逆多重化コンポーネント108)をメディアコンポーネントデータベース101から検索する(S1060)。ここでもS1030での処理と同様に図4の処理を行い、対象とするデータを逆多重化するのに適したコンポーネントを選択する。

#### 【0048】

メディア制御部106は、読み込んだ逆多重化コンポーネント108を用い、ファイル読込コンポーネント107からマルチメディアデータを読み込み、逆多重化を行う(S1070)。

#### 【0049】

メディア制御部106は、逆多重化されたメディアデータに付随するメディア情報（映像）を取得し、その映像を表示するためのコンポーネント（映像出力コンポーネント109）をメディア判定部102へ要求する（S1080）。

#### 【0050】

ここでいうメディア情報は、具体的には例えば図7に示すような形式のものである。メディア情報には、図7に示すようにメディアタイプ、メディアサブタイプ等のアイテムとそれに対応する値や文字列とが記載されている。図7において、“None”は該当するアイテムについて記載がないことを示している。

#### 【0051】

メディア判定部102は、メディア制御部106から入力されたメディア情報をもとに、映像を表示するためのコンポーネント（映像出力コンポーネント109）をメディアコンポーネントデータベース101から検索する（S1090）。ここでもS1030での処理と同様に図4の処理を行い、対象とするデータを映像出力するのに適したコンポーネントを選択する。

#### 【0052】

この場合のメディアタイプ判定関数40-1～Nの動作を示すフロー図を図8に示す。

#### 【0053】

まず、メディアタイプ判定関数40-1～Nは、上流のコンポーネントの出力から図7に示すメディア情報を取得する（S3010）。ここで、上流のコンポーネントとは、1つ前の処理が行われたコンポーネントを指す。したがって、ここでは、逆多重化コンポーネント108が上流のコンポーネントに該当する。処理は、メディアタイプ判定関数40-1～Nに対して適用され、適したコンポーネントが見つかるまで繰り返される。以下、メディアタイプ判定関数40-1についての動作について説明する。

#### 【0054】

メディアタイプ判定関数40-1は、読み込んだメディア情報のアイテムと、予め保持しているテーブルとを比較し、各アイテムがメディアタイプ判定関数の保持するテーブルの範囲内にあるか否かを判定する（S3020）。

#### 【0055】

図9は、メディアタイプ判定関数40-1が保持しているテーブルの一例を示したものである。メディアタイプ判定関数40-1は、各アイテムについて、順番に比較していくが、メディアタイプ判定関数40-1の保持しているテーブルの値の範囲内に無い場合は、その時点で判定は“NG”となる（S3050）。逆に範囲内に入っている場合は、まだメディアタイプ判定関数40-1が判定していないアイテムがあるか否かを判断する（S3030）。まだ全てのアイテムを判断していない場合は、S3020に戻る。全てのアイテムについて判定した場合は、判定は“OK”となる（S3040）。

#### 【0056】

なお、図9中の幾つかのアイテムについては“unspecified”（不特定の任意の値で良い）となっているが、このようなアイテムについては比較する必要がないので、範囲内に入っているか否かの判定を行わないようにすることもできる。

#### 【0057】

以上がメディアタイプ判定関数40-1の動作である。同様の動作を各メディアタイプ判定関数40-2～Nが行うことにより、処理に適したコンポーネントを選択することができる。

#### 【0058】

図3のフロー図を参照して、メディア制御部106は、読み込んだ映像出力コンポーネント109の出力先を映像表示部111へ接続する（S1100）。

#### 【0059】

そして、メディア制御部106は、逆多重化されたメディアデータのメディア情報（音声）を取得し、その音声を出力するためのコンポーネント（音声出力コンポーネント110）をメディア判定部102へ要求する（S1110）。

**【0060】**

メディア判定部102は、メディア制御部106から入力されたメディア情報をもとに、音声を出力するためのコンポーネント（音声出力コンポーネント110）をメディアコンポーネントデータベース101から検索する（S1120）。ここでもS1030での処理と同様に図4の処理を行い、対象とするデータを音声出力するのに適したコンポーネントを選択する。

**【0061】**

メディア制御部106は、読み込んだ音声出力コンポーネント110の出力先を音声出力部112へ接続する（S1130）。

**【0062】**

メディア制御部106は、各コンポーネントへ再生開始を要求する（S1140）。

**【0063】**

映像表示部111、音声出力部112において、映像、音声が出力される（S1150）。

**【0064】**

以上のように、メディアコンポーネントを検索する際、メディアタイプ関数40-1～Nを用いることによって、メディアコンポーネント全体を読み出すことなく、そのコンポーネントが処理に適しているか否かの判定を行うことができ、メディアコンポーネントの検索の高速化を図ることができる。

**【0065】**

また、メディアの判定のための方法が追加や変更された場合でも、コンポーネント内のメディアタイプ判定関数40-1～Nのみの変更で済み、データベース自体の再構築が不要となる。

**【0066】**

以上のように、本実施の形態によれば、コンポーネント内にコンポーネント本体とは別に、対象とするデータの処理に適するコンポーネントであるか否かを判定するタイプ判定関数を有し、データ処理装置には、読み込んだファイルまたは前段のコンポーネントにより処理され出力されたデータの情報を前記タイプ判定関数に適用する判定手段を備え、各コンポーネントと対応するタイプ判定を実施することで、判定のための属性が変更または追加となったとしても、タイプ判定関数のみを変更すれば足り、コンポーネント本体（従来のコンテキスト保持部全体に相当）を更新する必要はないため、効率的にデータベースを更新することが可能となる。

**【0067】**

また、メディアタイプ判定関数エントリを利用することによって、データベースよりコンポーネント全てを取得することなく、メディアタイプ判定関数エントリの参照先であるタイプ判定関数のみを読み込めばコンポーネントの評価を行うことができ、目的のコンポーネントを取得するまでの時間を短縮することができる。

**【0068】**

（実施の形態2）

図10は、本発明の実施の形態2におけるデータ処理装置200の構成を示すブロック図である。なお、本実施の形態のデータ処理装置200は、実施の形態1で説明したデータ処理装置と同様の基本的構成を有しており、同一の構成要素には同一の参照符号を付し、その詳細な説明を省略する。

**【0069】**

データ処理装置200におけるデータ処理装置100との相違点は、過去にこのデータ処理装置200で組み立てたメディアコンポーネントの構成パターンを格納するためのコンポーネント構成データベース201と、ファイルシステム104から読み出されたデータに適用可能なメディアコンポーネントの構成パターンをコンポーネント構成データベース201から検索する構成パターン検索部202と、必要なメディアコンポーネントがデータ処理装置200の内部に保持されていない場合に、ネットワーク210を介して接続

されるメディアコンポーネントデータベースサーバ220から必要なメディアコンポーネントをダウンロードするダウンロード部203と、メディアコンポーネントの配置に成功した構成パターンをコンポーネント構成データベース201に登録する構成パターン登録部204と、が追加された点、並びに、メディア判定部102、操作入力部103及びメディア処理部105の代わりにメディア判定部205、入力部206及びメディア処理部207を有する点である。また、メディア処理部207は、メディア処理部105におけるメディア制御部106の代わりにメディア制御部208を有する。

#### 【0070】

本実施の形態の主な特徴は、過去に組み立てたメディアコンポーネントの構成パターンをコンポーネント構成データベース201に格納しておき、処理対象のファイル（データ）の情報に基づいて、格納された構成パターンの中から処理に適した構成パターンを検索することである。

#### 【0071】

図11は、コンポーネント構成データベース201にて構成パターンを保持するための保持テーブルの一例を示す図である。

#### 【0072】

この保持テーブルにおいて、「構成パターン」は、構成パターン名を示す。「判定関数」は、メディア情報に対応する構成パターンを検索するときに利用される判定関数を示す。「コンポーネント数」は、処理に必要なコンポーネント数を示す（この例では、5個）。「コンポーネントID」は、構成パターンに含まれる各コンポーネントの識別子を示し、各コンポーネントを一意に特定するために用いられる。「コンポーネント格納先」は、各コンポーネントの実体が、メディアコンポーネントデータベース101のどの位置（アドレス）に格納されているかを示す。「コンポーネント所在」は、そのコンポーネントの実体が格納されているメディアコンポーネントデータベースサーバ220のアドレスを示している。「コンポーネント接続数」は、各コンポーネント間を接続する配線の数を示している。「接続方法」は、各コンポーネントの接続方法を示している。例えば {0、1} は、構成パターン中の0番目のコンポーネント（図11のコンポーネント構成例ではID [0] と記載）及び1番目のコンポーネント（図11のコンポーネント構成例ではID [1] と記載）が接続されることを示している。

#### 【0073】

以下、上記構成を有するデータ処理装置200における動作について説明する。まず、メディアコンポーネントの構成パターンの登録について説明する。図12は、構成パターンの登録動作手順を説明するためのフローチャートである。

#### 【0074】

まず、メディア制御部208は、メディアコンポーネントの配置に成功したとき（つまり、読み込んだファイルを適切に処理することができたとき）、そのメディアコンポーネントの構成パターンを構成パターン登録部204に通知する（S4000）。

#### 【0075】

そして、構成パターン登録部204は、通知された構成パターンが既にコンポーネント構成データベース201に登録されているか否かを調べ（S4100）、既に登録されている場合は何も行わず、登録されていない場合はその構成パターンを保持するための保持テーブルを作成し、コンポーネント構成データベース201に登録する（S4200）。

#### 【0076】

次いで、データ処理装置200でのマルチメディアファイルの再生について説明する。図13は、データ処理装置200でのマルチメディアファイルの再生動作手順を説明するためのフローチャートである。

#### 【0077】

まず、構成パターン検索部202は、今回の処理に適した構成パターンがコンポーネント構成データベース201に存在しているか否かを検索し（S5000）、その構成パターンがコンポーネント構成データベース201に存在している場合は（S5100：Y E

S) ステップ S5200 に進み、その構成パターンがコンポーネント構成データベース 201 に存在していない場合は (S5100:NO) ステップ S5300 に進む。

【0078】

ステップ S5200 において、メディア判定部 205 は、構成パターンに含まれるメディアコンポーネントの実体をメディアコンポーネントデータベース 101 から読み出す。一方、ステップ S5300 においては、メディア判定部 205 は、必要なコンポーネントの実体を上流側から順にメディアコンポーネントデータベース 101 から読み出す、すなわち、実施の形態 1 と同様の動作を行うこととなる。

【0079】

そして、メディア制御部 208 は、読み出されたメディアコンポーネントを接続する (S5400)。

【0080】

ここで、上記のステップ S5000 の詳細な動作について説明する。ここでは図 14 及び図 15 を参照し二つの動作例を挙げる。図 14 に示す動作例では、入力部 206 が構成パターンを指定し (S5010)、構成パターン検索部 202 が、指定された構成パターンに対応する構成パターン (つまり指定された構成パターンと同一のもの) を検索する (S5020)。また、図 15 に示す動作例では、メディアタイプ判定関数が上流のコンポーネントのメディア情報を取得し (S5030)、構成パターン検索部 202 が、取得されたメディア情報に対応する構成パターンを、コンポーネント構成データベース 201 の判定関数を利用して検索する (S5040)。図 15 に示す動作は、例えば、図 14 について説明した構成パターン指定がなかったときに行われる。

【0081】

上記のステップ S5010 での構成パターン指定方法の例としては、ユーザから入力部 206 への入力によりアプリケーションが起動されたときに、そのアプリケーションに対応する構成パターンを構成パターン検索部 202 に通知する方法が挙げられる。

【0082】

また、起動されるアプリケーションの例としては、カメラアプリが挙げられる。図 16 は、カメラアプリ起動中の操作と、その操作に応じて映像表示部 111 に表示される画面との例を説明するための図である。

【0083】

まず、カメラアプリが起動されると、入力部 206 は、カメラアプリに適したメディアコンポーネントの構成パターンを構成パターン検索部 202 に通知する。そして、構成パターン検索部 202 が通知された構成パターンに対応する構成パターン (例えば、図 11 の構成パターン「Camera\_Record」) を検索する。そして、この構成パターンに含まれるメディアコンポーネントが入力部 206 によって読み出されメディア制御部 208 によって接続される。これにより、内蔵のカメラ (不図示) から入力される映像が映像表示部 111 に表示される。その後、記録開始/停止用のボタン 206a が押下されると、カメラからの映像の記録が開始される。さらに、ボタン 206a が再度押下されると、入力部 206 がこれを検出し、映像の記録が停止される。

【0084】

また、機能ボタン 206c を押下すると、入力部 206 がこれを検出し、機能メニューが映像表示部 111 に表示され、例えば、色調切替、外部出力設定、画面サイズ変更及び記録形式変更等の設定機能を実行することが可能になる。例えば色調切替は、設定用のボタン 206b の右側部を押下することにより映像の色調を明るくし、ボタン 206b の左側部を押下することにより映像の色調を暗くする機能である。また、例えば画面サイズ変更は、ボタン 206b の上側部を押下することにより画面サイズを拡大し、ボタン 206b の下側部を押下することにより画面サイズを縮小する機能である。

【0085】

上記の各設定機能を実現するために、各設定機能に適したメディアコンポーネントの構成パターンが予め決められている。いずれかの設定機能の実行がユーザによって指示され

たときは、指示された設定機能に対応する構成パターンの検索、当該構成パターンに含まれるメディアコンポーネントの読み出し及び読み出されたメディアコンポーネントの接続が行われ、指示された設定機能が実行される。そして、設定が完了したとき及び設定がキャンセルされたとき、機能メニューの表示が終了する。また、終了ボタン 206d を押下すると、入力部 206 がこれを検出し、カメラアプリが終了する。

#### 【0086】

次いで、図 13 のステップ S5200 の動作の詳細について説明する。図 17 は、構成パターンに含まれるメディアコンポーネントを読み出す手順を説明するためのフローチャートである。

#### 【0087】

まず、メディア判定部 205 は、該当する構成パターンの保持テーブルに記載されたコンポーネント格納先のアドレスを参照する (S6000)。そして、コンポーネント格納先が “None” ではない (例えば、図 11 のアドレス {0x0200}) 場合 (S6100:NO)、メディアコンポーネントデータベース 101 内の上記アドレスからコンポーネントを読み出す (S6200)。一方、コンポーネント格納先が “None” の場合 (S6100:YES)、メディア判定部 205 は、保持テーブルに記載されたコンポーネント所在 (例えば、図 11 のアドレス {http://hoge.hoge.hoge/cam\_cap}) を参照してダウンロード部 203 に通知し、ダウンロード部 203 がネットワーク 210 を介して接続されるメディアコンポーネントデータベースサーバ 220 の上記コンポーネント所在から該当するメディアコンポーネントをダウンロードし、メディア判定部 205 がダウンロード部 203 からメディアコンポーネントを読み出す (S6300)。

#### 【0088】

なお、コンポーネント格納先を “None” にするか否かは、例えばメディアコンポーネントの利用頻度に応じて決定される。

#### 【0089】

そして、読み出したメディアコンポーネントの数が、保持テーブルに記載されたコンポーネント数に達したか否かを判定し (S6400)、コンポーネント数に達した場合は手順を終了し、コンポーネント数に達していない場合はステップ S6000 に戻り上記の手順を繰り返す。

#### 【0090】

図 18 は、図 17 のステップ S6200 の動作、つまりメディアコンポーネントデータベース 201 からのメディアコンポーネント読み出しを詳細に示したフローチャートである。

#### 【0091】

まず、保持テーブルに記載されたコンポーネント格納先に基づいて、メディアコンポーネントデータベース 101 から該当するメディアコンポーネントを読み出す (S6210)。そして、読み出しに成功した場合 (S6220:YES)、処理を終了する。一方、読み出しに失敗した場合は (S6220:NO)、メディア判定部 205 は、ダウンロード部 203 にネットワーク 210 を介したメディアコンポーネントデータベースサーバ 220 から該当するメディアコンポーネントをダウンロードする (S6230)。

#### 【0092】

図 19 は、図 17 のステップ S6300 の動作、つまりダウンロード部 203 からのメディアコンポーネント読み出しを詳細に示したフローチャートである。

#### 【0093】

まず、メディア判定部 205 は、ネットワーク 210 を介しメディアコンポーネントデータベースサーバ 220 からメディアコンポーネントをダウンロード部 203 にダウンロードし、ダウンロードされたメディアコンポーネントをダウンロード部 203 から読み出す (S6310)。そして、読み出しに成功した場合 (S6320:YES)、メディア判定部 205 は、ダウンロードしたメディアコンポーネントをメディアコンポーネントデータベースに登録し (S6330)、構成パターン検索部 202 は、保持テーブルに記載

されたコンポーネント格納先を更新する(S6340)。一方、読み出しに失敗した場合は(S6320:NO)、メディアコンポーネントの読み出しが行えない旨のエラー処理を行う(S6350)。

#### 【0094】

このように、本実施の形態によれば、過去に成功した構成パターンをコンポーネント構成データベース201に格納しておき、当該構成パターンと同一の処理が必要になったときに当該構成パターンをコンポーネント構成データベース201から検索し、検索された構成パターンに含まれるメディアコンポーネントのみを読み出すため、目的のメディアコンポーネントを取得するまでの時間を従来より大幅に短縮することができる。

#### 【0095】

また、データ処理装置200の外部と接続可能なダウンロード部203を備えることで、ある処理を実現するために必要なメディアコンポーネントがデータ処理装置200の内部に記憶されていないときに、外部のメディアコンポーネントデータベースサーバ220からネットワーク210を介して当該メディアコンポーネントをダウンロードするため、例えば新規のメディアコンポーネントを取得する必要がある場合でもメディアコンポーネントデータベースサーバ220から即座にダウンロードすることが可能になるだけでなく、例えば利用頻度の低いメディアコンポーネントをメディアコンポーネントデータベース101に格納しておく必要がなくなり、メディアコンポーネントデータベース101のメモリを効率的に使用することが可能になる。

#### 【0096】

なお、本実施の形態では、ダウンロード部203は、メディアコンポーネントデータベースサーバ220からメディアコンポーネントを取得しているが、外部からのメディアコンポーネント取得の方法はこれに限定されない。例えば、コンポーネントを格納した記録媒体をデータ処理装置200に直接接続することができる場合は、その記録媒体から必要なコンポーネントをダウンロードしても良い。

#### 【0097】

また、本発明は上記の実施の形態に限定されず、様々な実施の形態を採ることが可能である。例えば、上記のデータ処理方法をコンピュータ上でソフトウェアとして実行しても良い。すなわち、上記の実施の形態で説明したデータ処理方法を実行するプログラムを予め例えばROM(Read Only Memory)等の記録媒体に記録しておき、そのプログラムをCPU(Central Processor Unit)によって動作させるようにしても良い。

#### 【産業上の利用可能性】

#### 【0098】

本発明に係るデータ処理装置、データ処理方法及びデータ処理プログラムは、新規コンポーネントが追加された場合でもデータベースの再構築を不要にする効果を有し、ある機能を実現するために組み合わされたソフトウェア部品から構成されるコンポーネントを読み込んで処理を実行するのに有用である。

#### 【図面の簡単な説明】

#### 【0099】

【図1】 本発明の実施の形態1におけるデータ処理装置の構成を示すブロック図

【図2】 本発明の実施の形態1におけるメディアコンポーネントデータベースの構成を示す図

【図3】 本発明の実施の形態1におけるマルチメディアファイル再生手順を示すフロー図

【図4】 本発明の実施の形態1におけるメディア判定部の処理手順を示すフロー図

【図5】 本発明の実施の形態1における多重化ファイルの構成を示す図

【図6】 本発明の実施の形態1における多重化情報の一例を示す図

【図7】 本発明の実施の形態1におけるメディア情報の一例を示す図

【図8】 本発明の実施の形態1におけるメディアタイプ判定関数の動作を示すフロー図

【図 9】本発明の実施の形態 1 におけるメディアタイプ判定関数が保持するテーブルの一例を示す図

【図 10】本発明の実施の形態 2 におけるデータ処理装置の構成を示すブロック図

【図 11】本発明の実施の形態 2 における保持テーブルの一例を示す図

【図 12】本発明の実施の形態 2 における構成パターンの登録動作手順を説明するためのフロー図

【図 13】本発明の実施の形態 2 におけるマルチメディアファイル再生動作手順を説明するためのフロー図

【図 14】本発明の実施の形態 2 における構成パターン検索動作の第 1 の例を説明するためのフロー図

【図 15】本発明の実施の形態 2 における構成パターン検索動作の第 2 の例を説明するためのフロー図

【図 16】本発明の実施の形態 2 における表示画面の例を説明するための図

【図 17】本発明の実施の形態 2 における構成パターンに含まれるメディアコンポーネントの読み出し手順を説明するためのフロー図

【図 18】本発明の実施の形態 2 におけるメディアコンポーネントデータベースからのメディアコンポーネント読み出し動作を説明するためのフロー図

【図 19】本発明の実施の形態 2 におけるダウンロード部からのメディアコンポーネント読み出し動作を説明するためのフロー図

【図 20】従来のデータ処理装置の構成を示すブロック図

【図 21】従来のデータ処理装置における検索式の一例を示す図

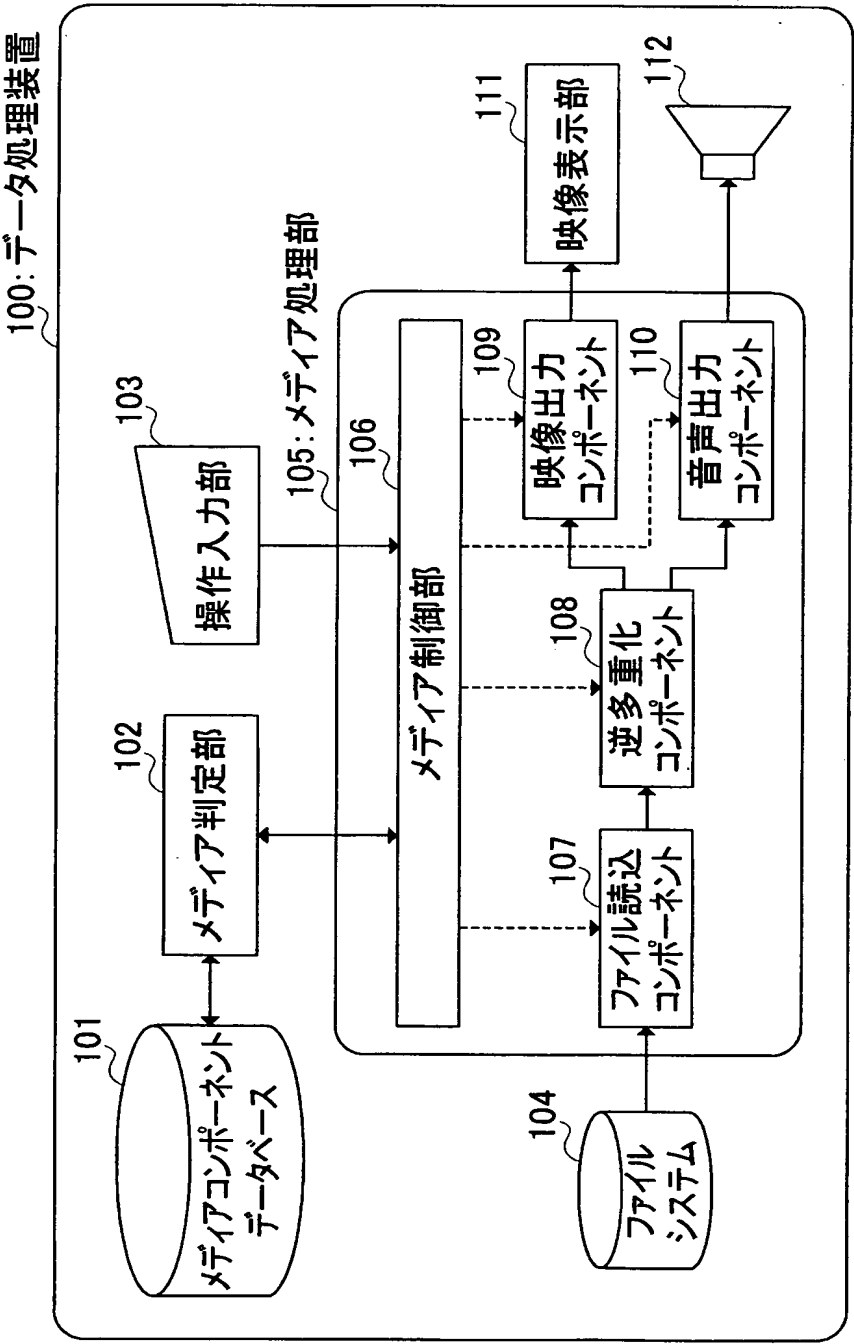
【符号の説明】

【0100】

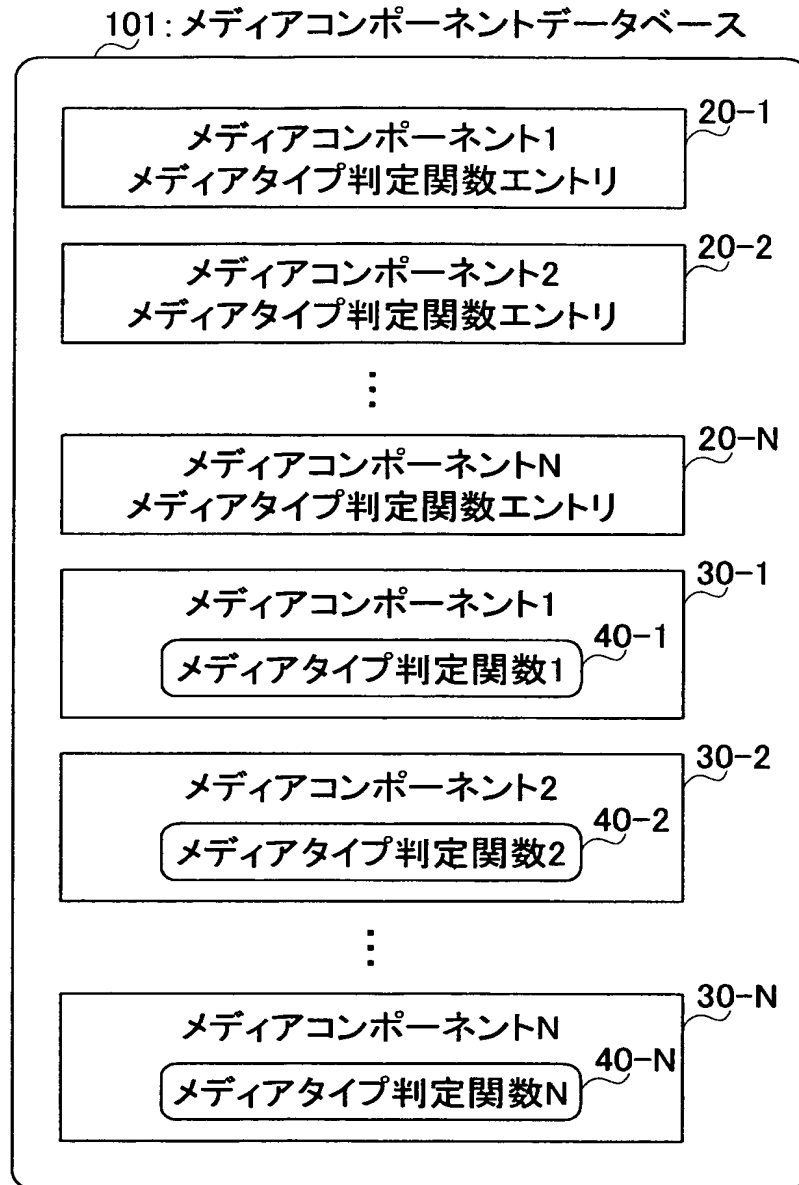
- 100、200 データ処理装置
- 101 メディアコンポーネントデータベース
- 102、205 メディア判定部
- 103 操作入力部
- 104 ファイルシステム
- 105、207 メディア処理部
- 106、208 メディア制御部
- 107 ファイル読込コンポーネント
- 108 逆多重化コンポーネント
- 109 映像出力コンポーネント
- 110 音声出力コンポーネント
- 111 映像表示部
- 112 音声出力部
- 20-1～N メディアタイプ判定関数エントリ
- 30-1～N メディアコンポーネント
- 40-1～N メディアタイプ判定関数
- 201 コンポーネント構成データベース
- 202 構成パターン検索部
- 203 ダウンロード部
- 204 構成パターン登録部
- 206 入力部
- 206a、206b ボタン
- 206c 機能ボタン
- 206d 終了ボタン
- 210 ネットワーク
- 220 メディアコンポーネントデータベースサーバ



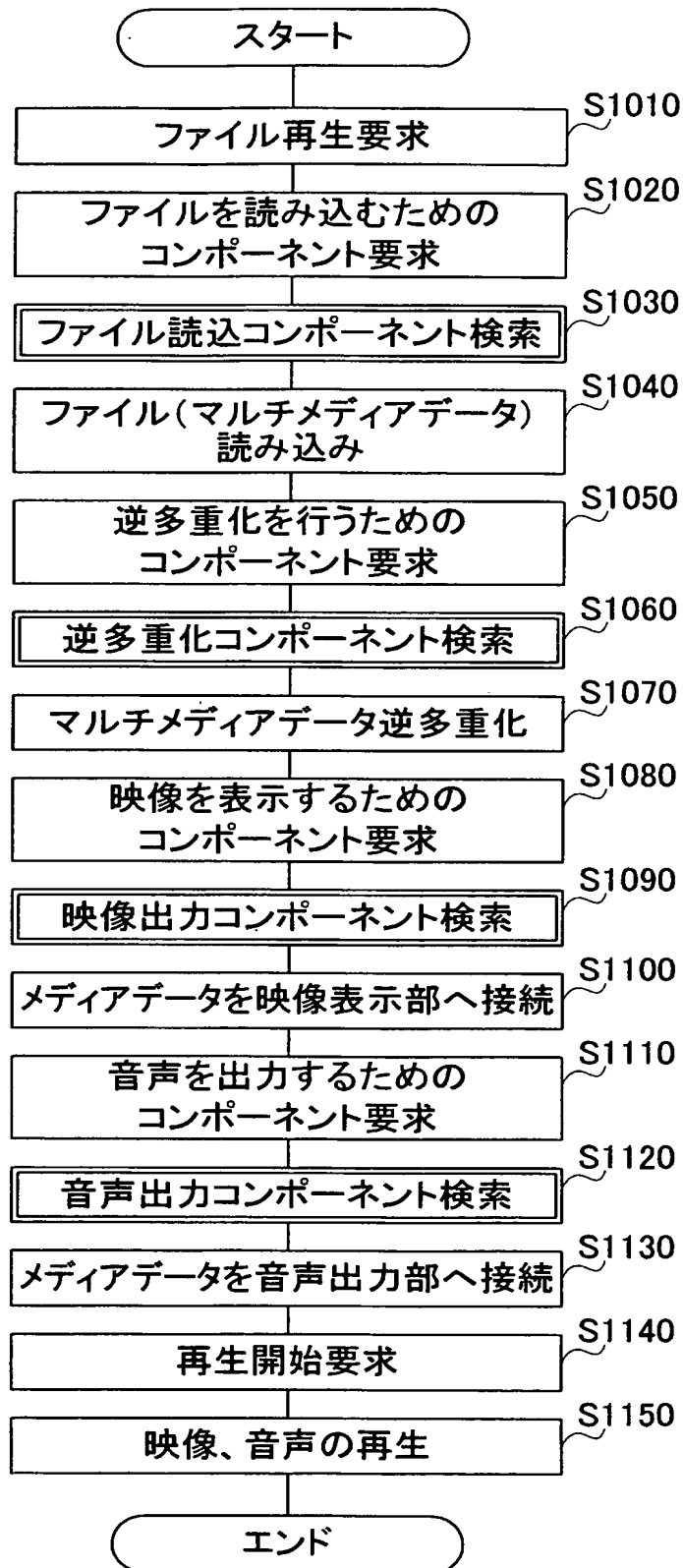
【書類名】 図面  
【図 1】



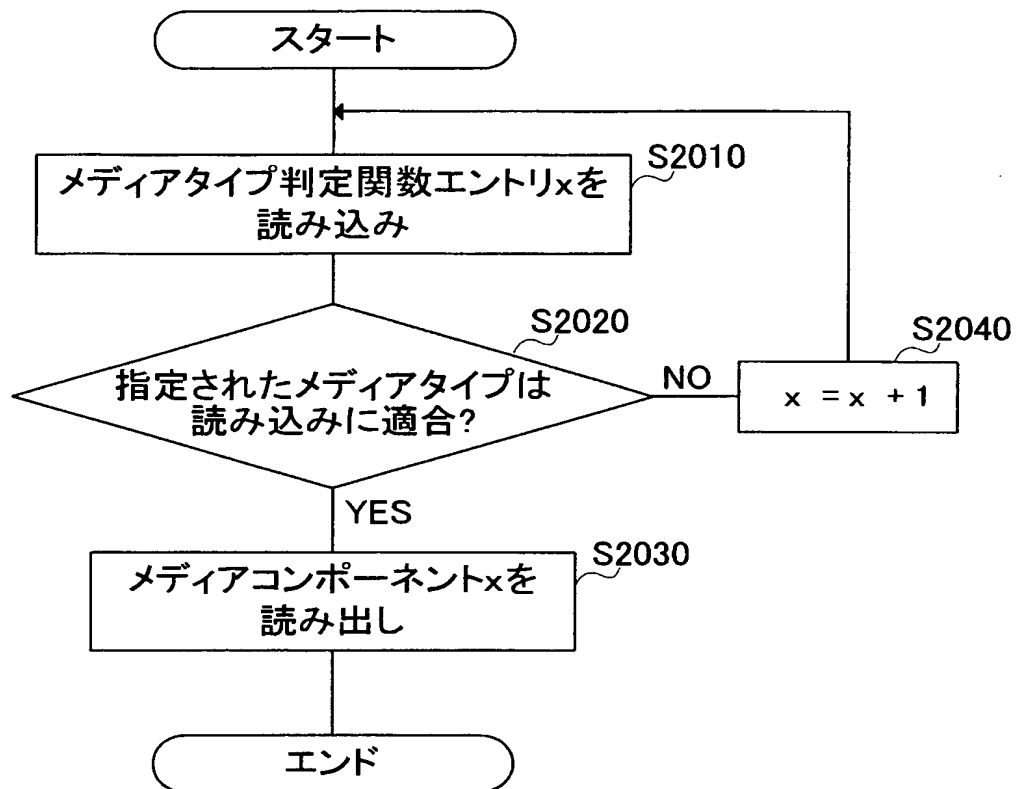
【図 2】



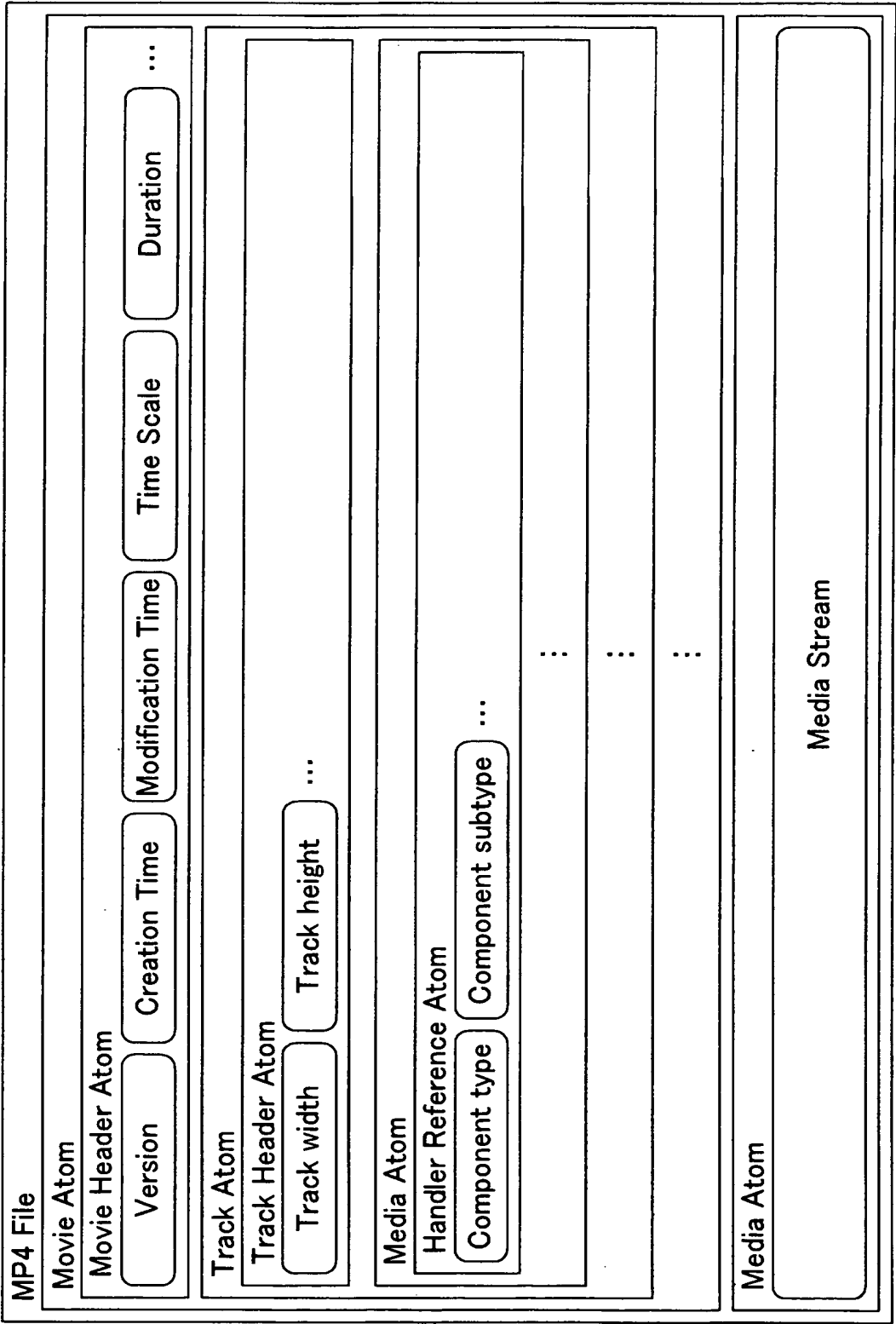
【図 3】



【図 4】



【図 5】



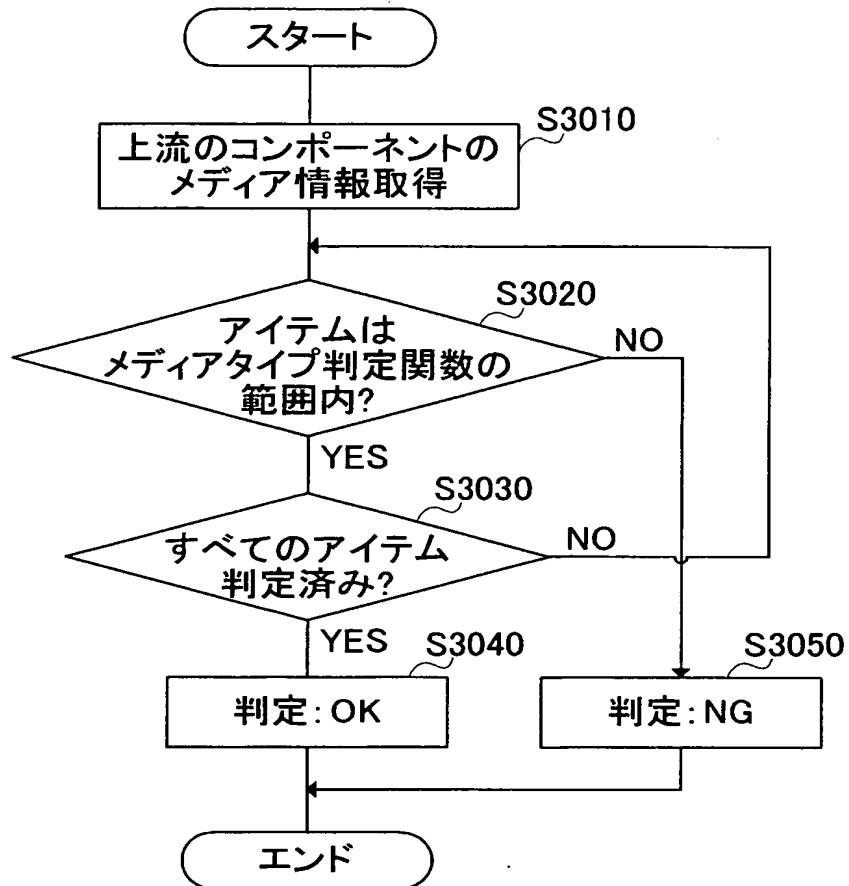
【図 6】

	MP4 Atom
メディアタイプ	Movie->Track->Media->Handler Reference(Component subtype)
メディアサブタイプ	Movie->Track->Media->Media Information->Sample Table ->Sample Description->Visual Sample Entry(Type)
コーデック	Movie->Track->Media->Media Information->Sample Table ->Sample Description->Visual Sample Entry(ESD)
バージョン	Movie->Track->Media->Media Information->Sample Table ->Sample Description->Visual Sample Entry(Decoder_version)
平均ビットレート	Movie->Track->Media->Media Information->Sample Table ->Sample Description->Visual Sample Entry(ESD)
最大ビットレート	Movie->Track->Media->Media Information->Sample Table ->Sample Description->Visual Sample Entry(ESD)
チャンネル数	None
サンプリング周波数	None
幅	Movie->Track->Media->Media Information->Sample Table ->Sample Description->Visual Sample Entry(Width)
高さ	Movie->Track->Media->Media Information->Sample Table ->Sample Description->Visual Sample Entry(Height)

【図 7】

アイテム	逆多重化後
メディアタイプ	Video
メディアサブタイプ	MPEG-4
コーデック	Simple Profile
バージョン	1.0
平均ビットレート	400
最大ビットレート	600
チャンネル数	None
サンプリング周波数	None
幅	176
高さ	144

【図 8】

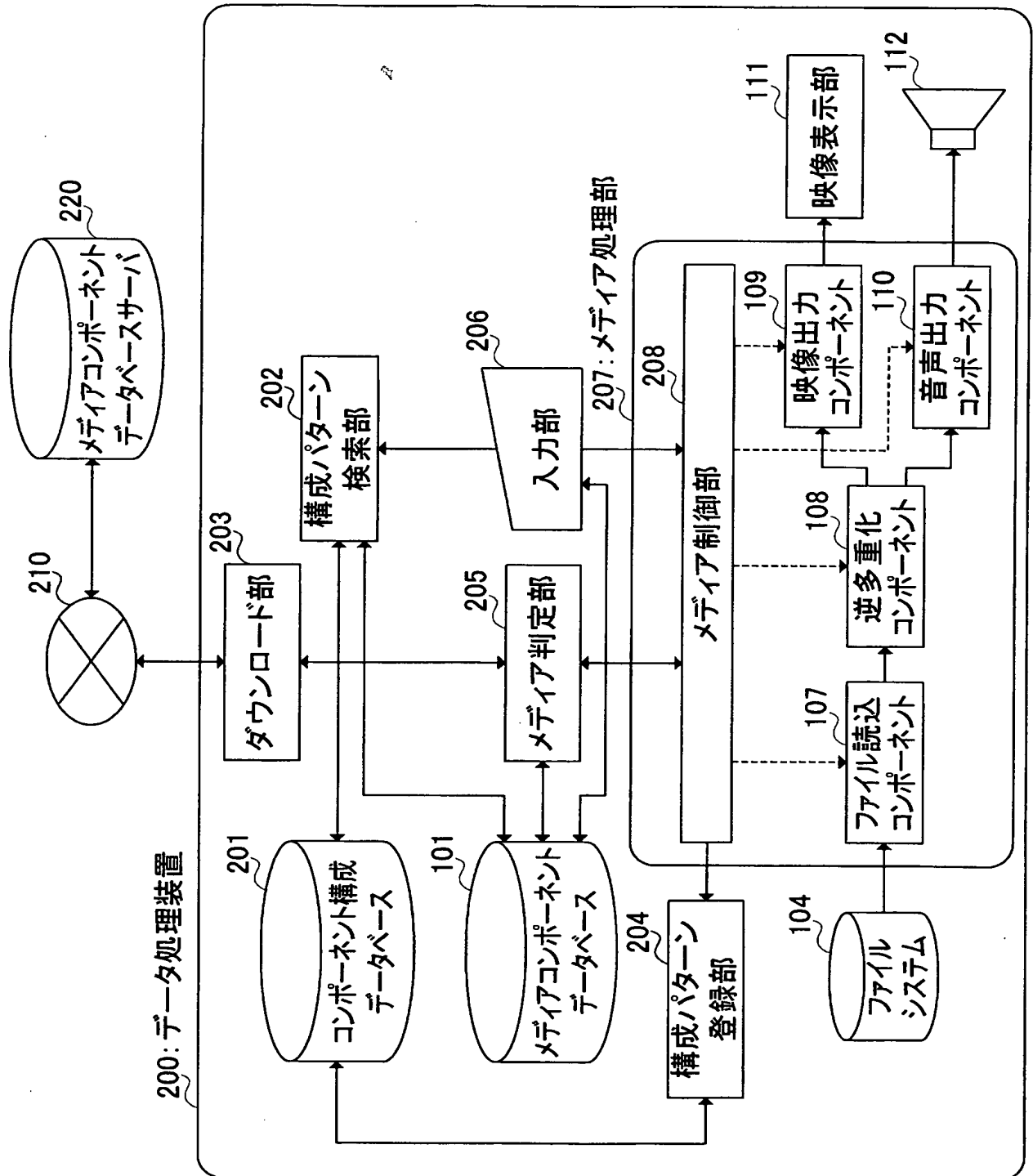




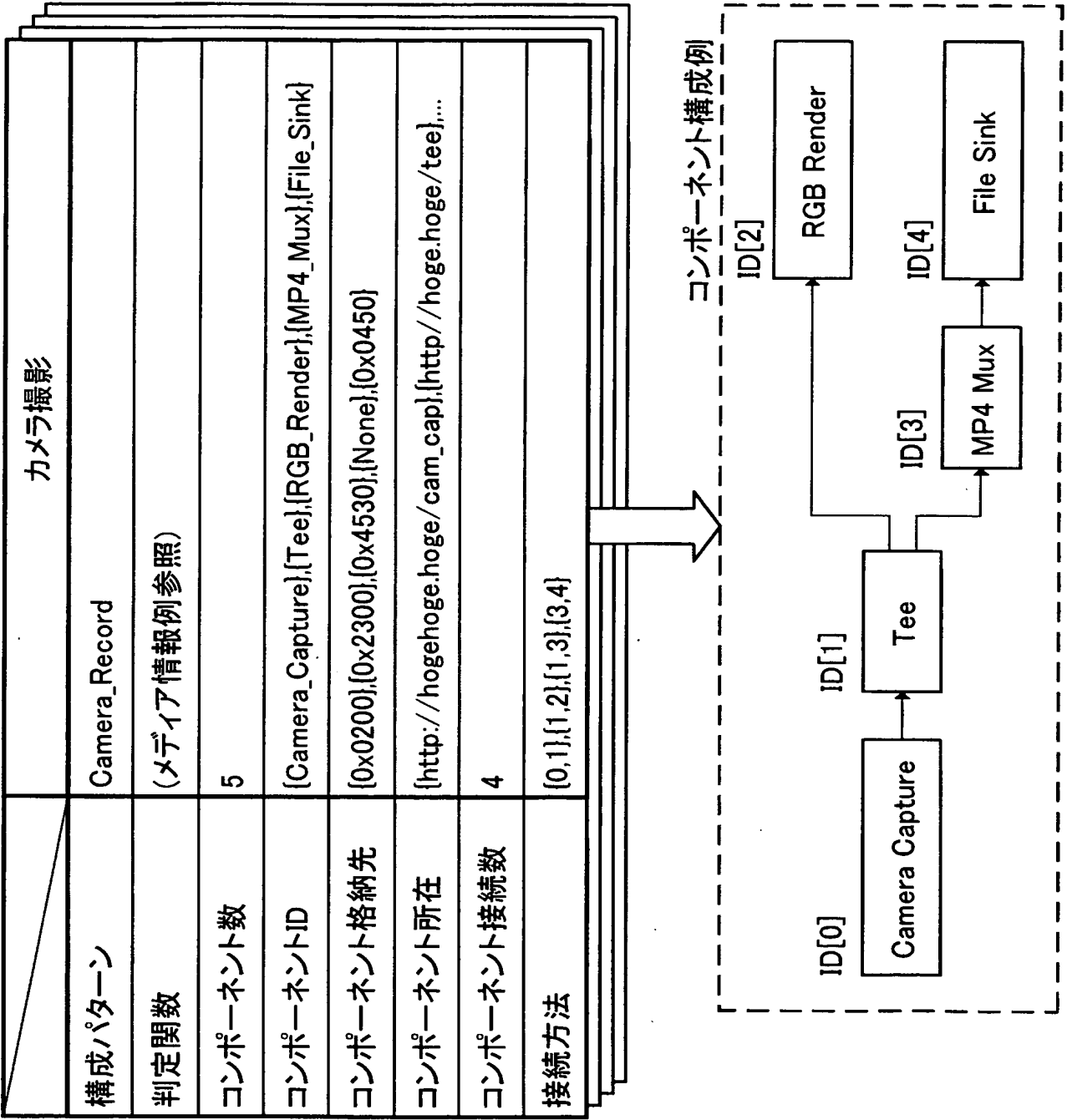
【図 9】

アイテム	メディアタイプ判定関数
メディアタイプ	Stream
メディアサブタイプ	MP4
コーデック	unspecified
バージョン	<=1.0
平均ビットレート	unspecified
最大ビットレート	unspecified
チャンネル数	unspecified
サンプリング周波数	unspecified
幅	unspecified
高さ	unspecified

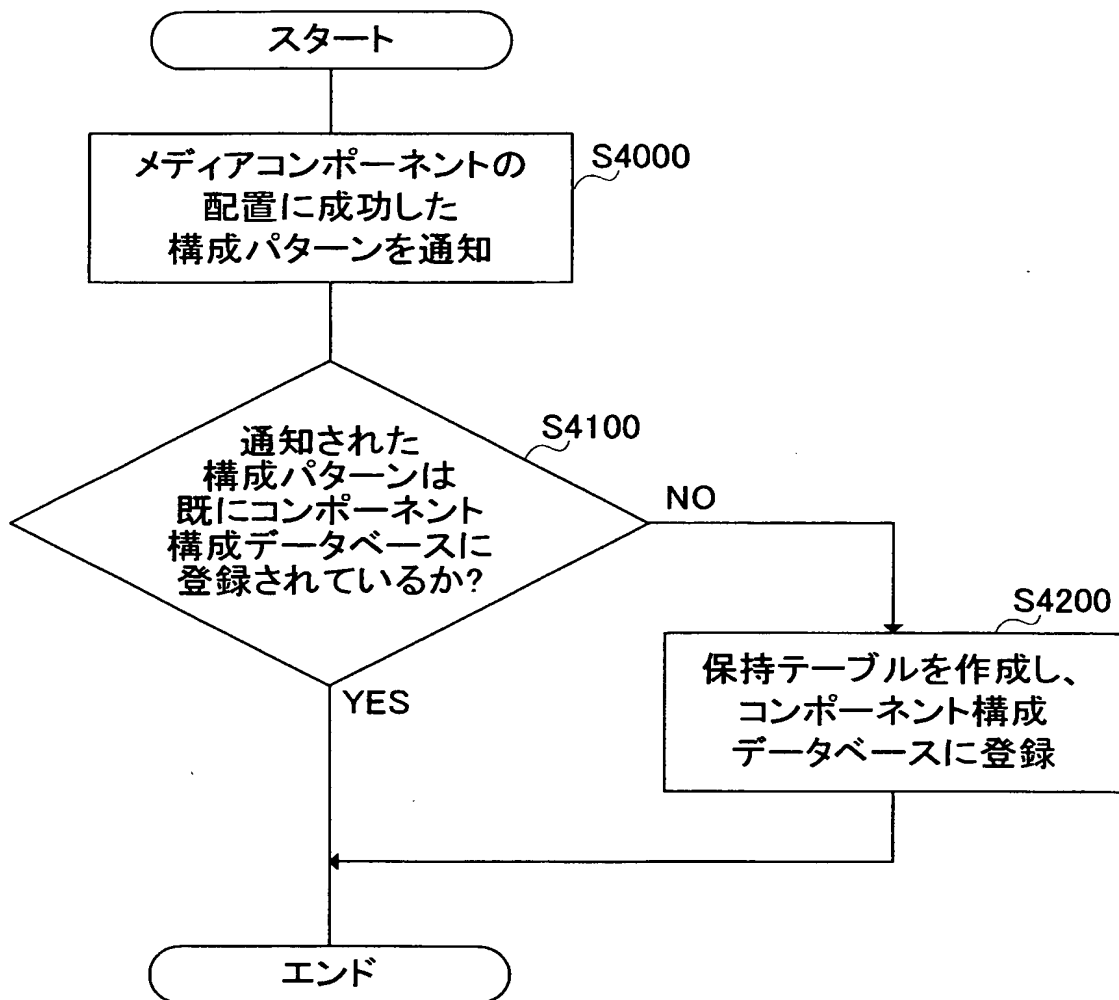
【図10】



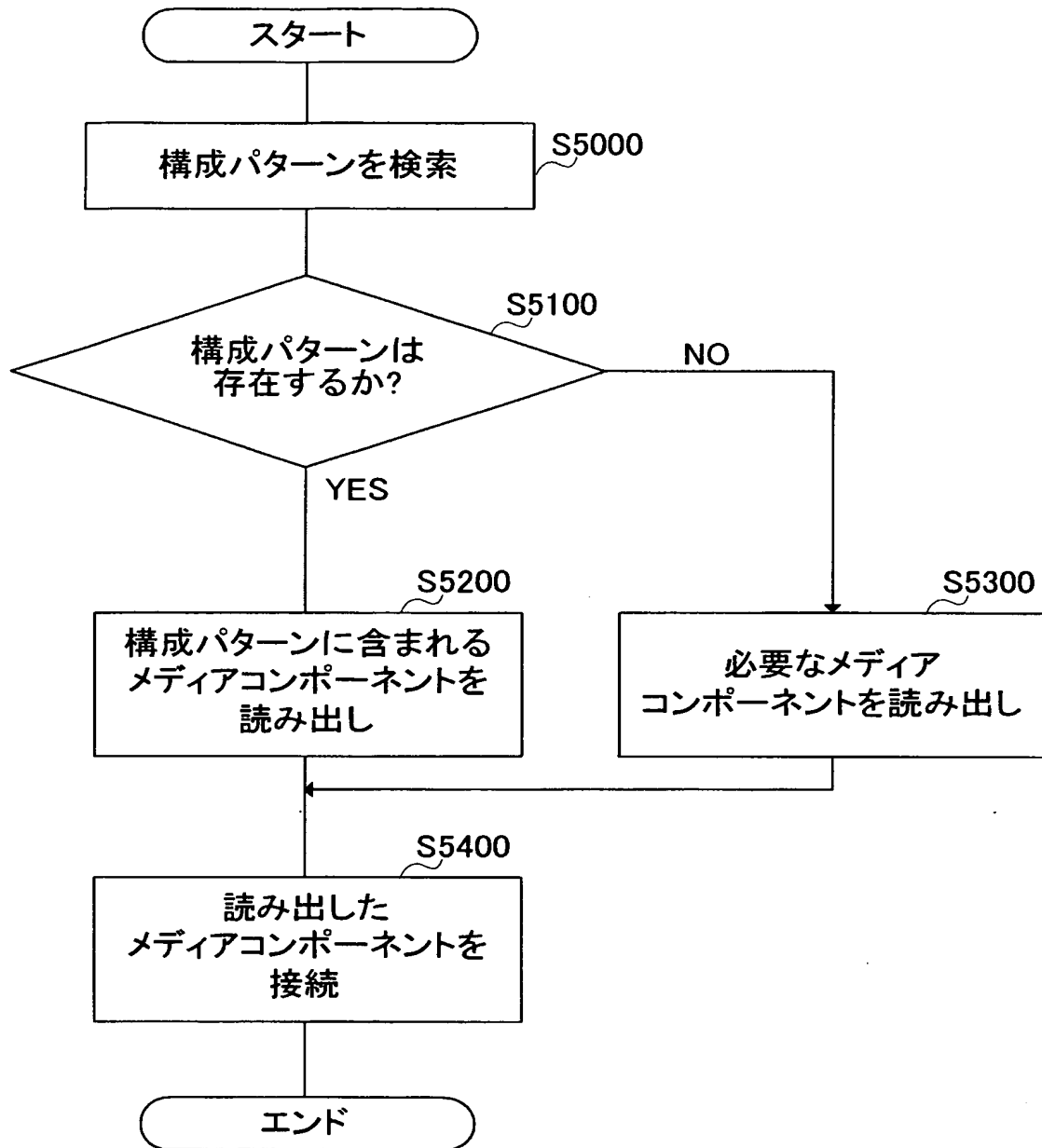
【図 11】



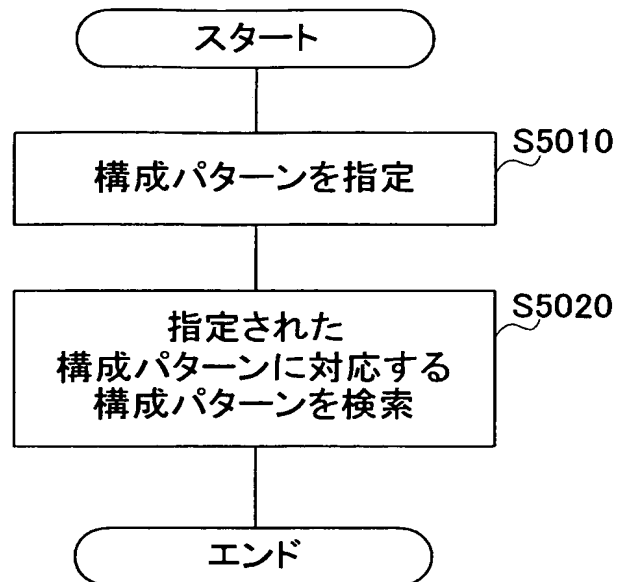
【図 12】



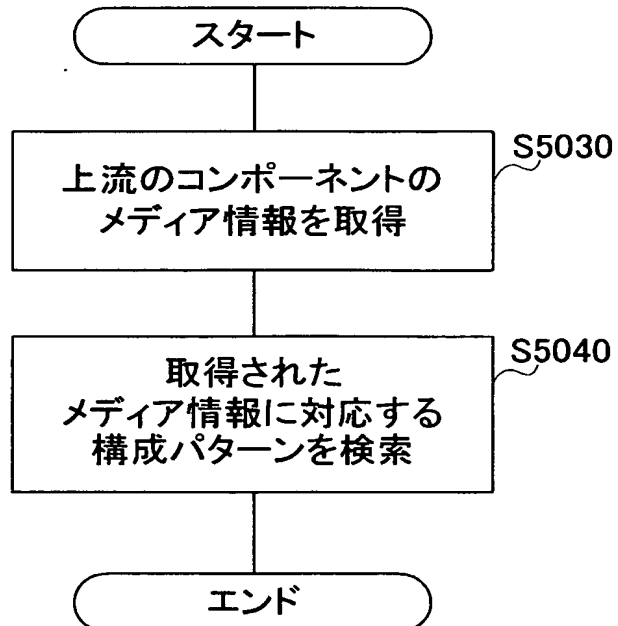
【図 13】



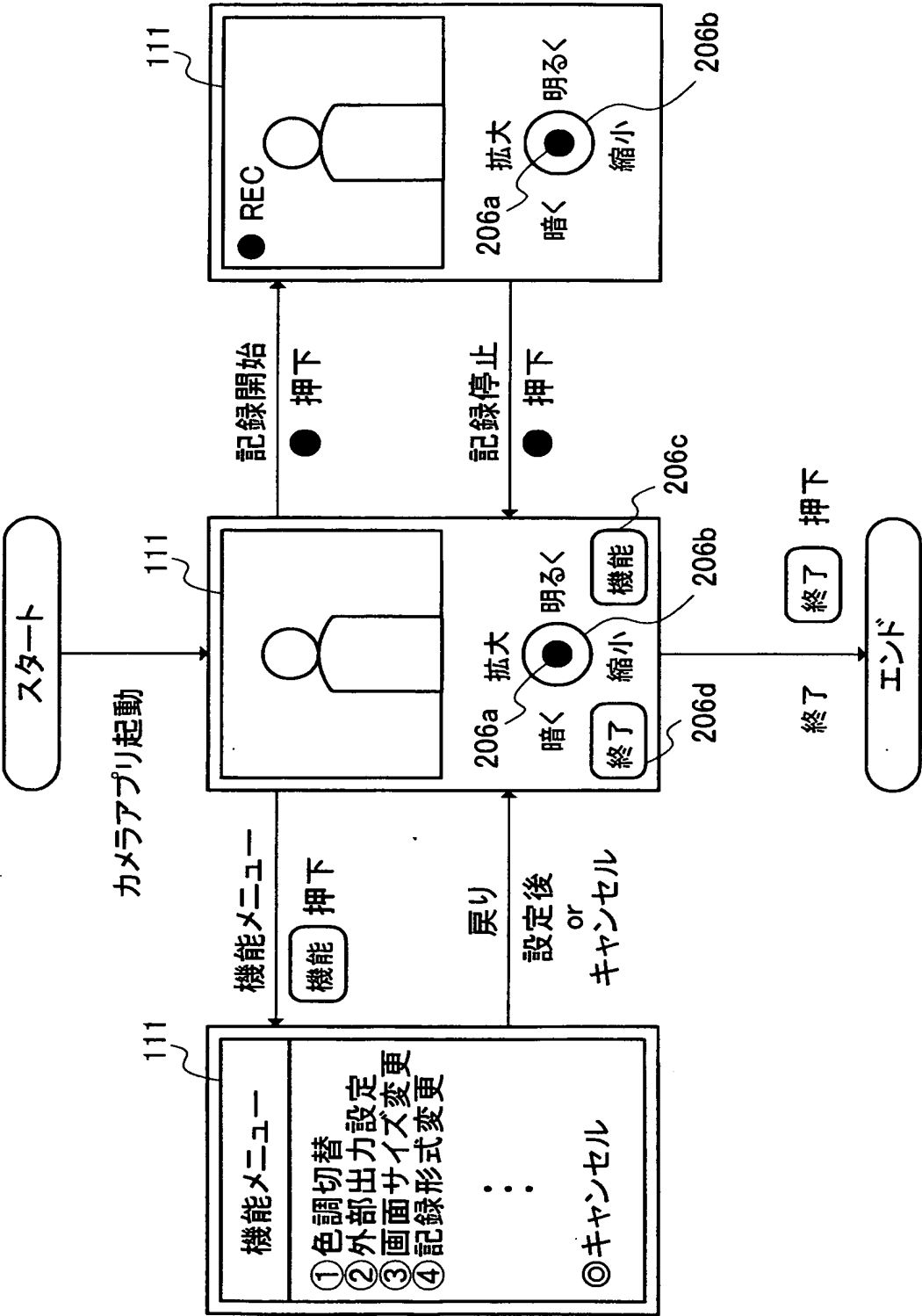
【図 14】



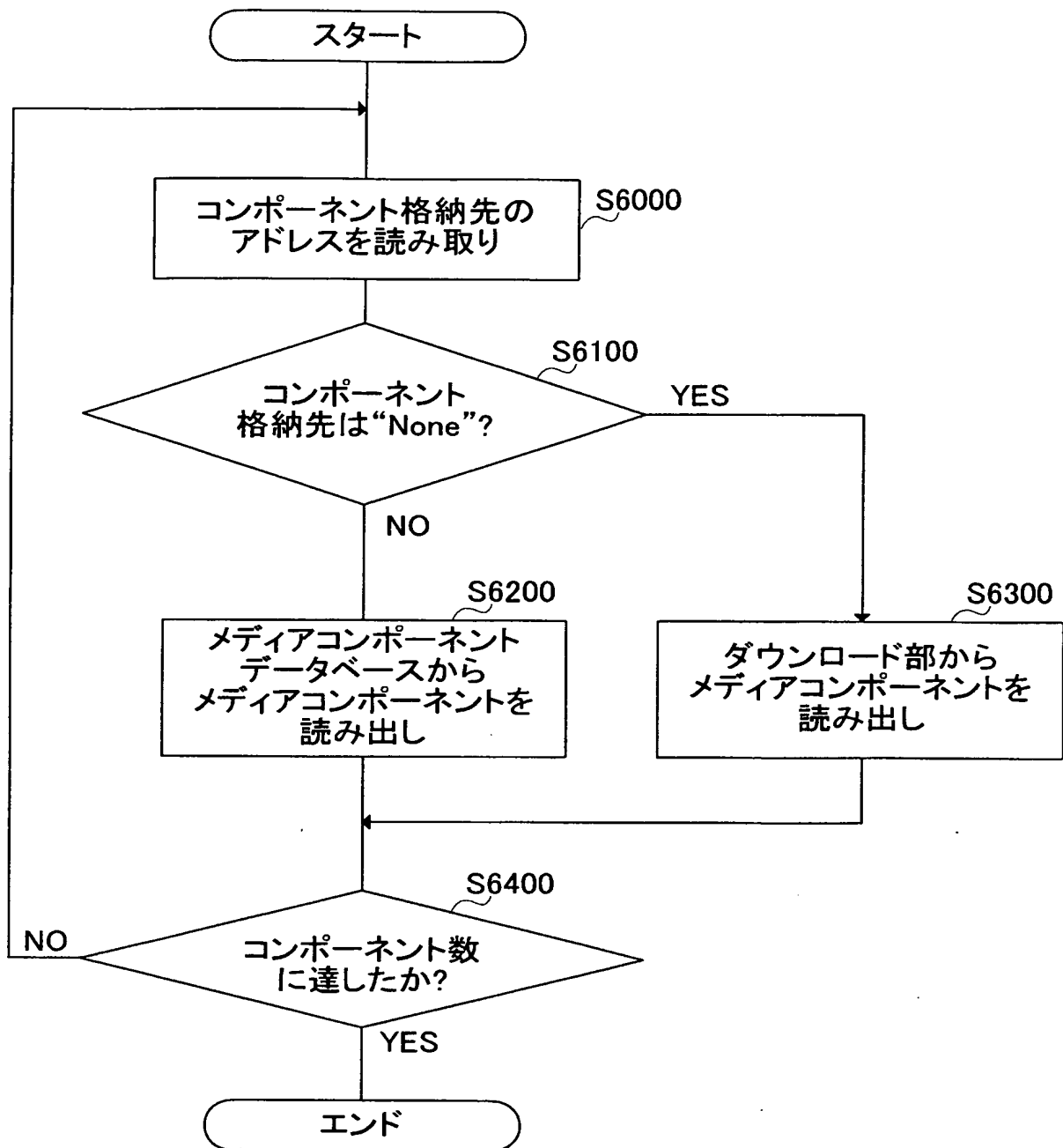
【図 15】



【図 16】

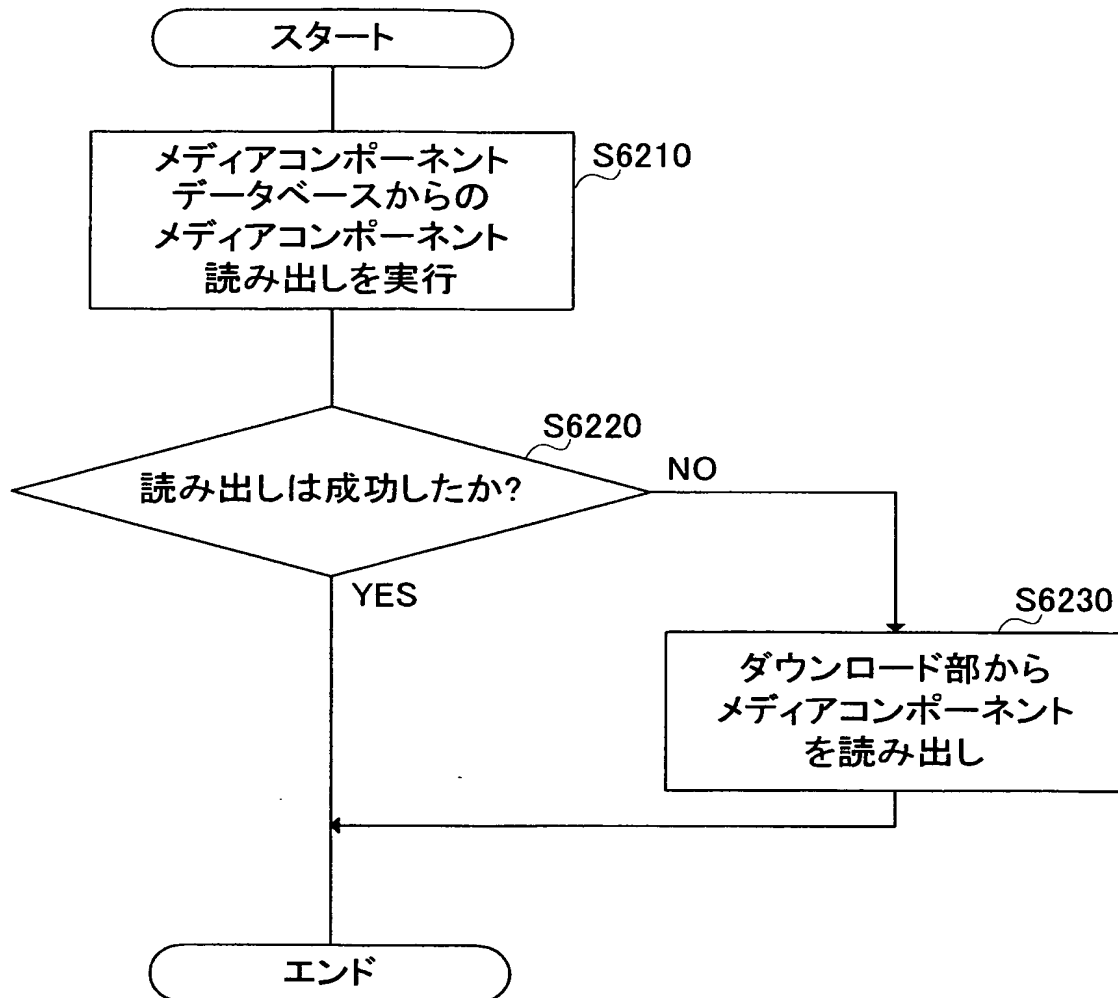


【図 17】

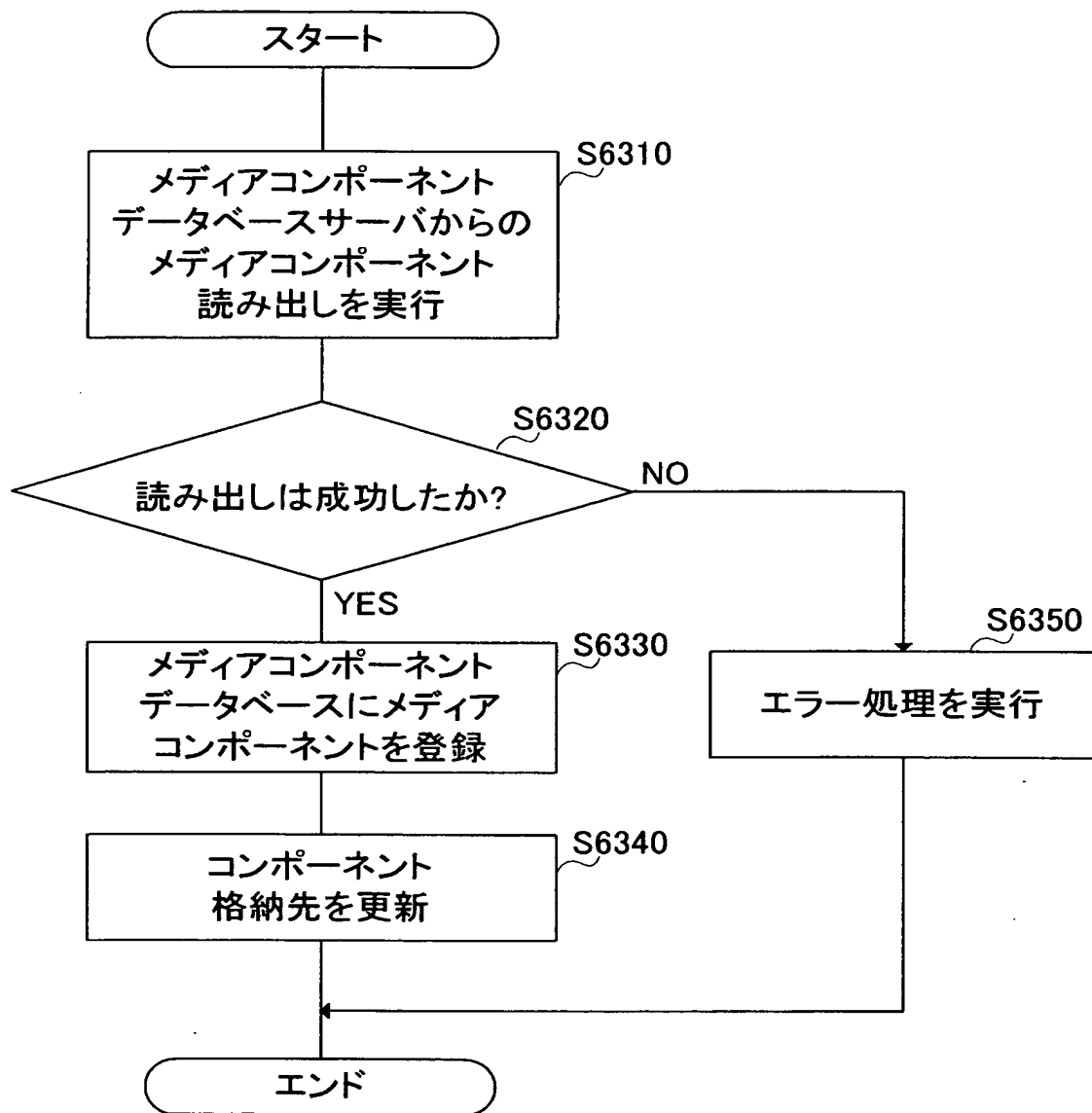




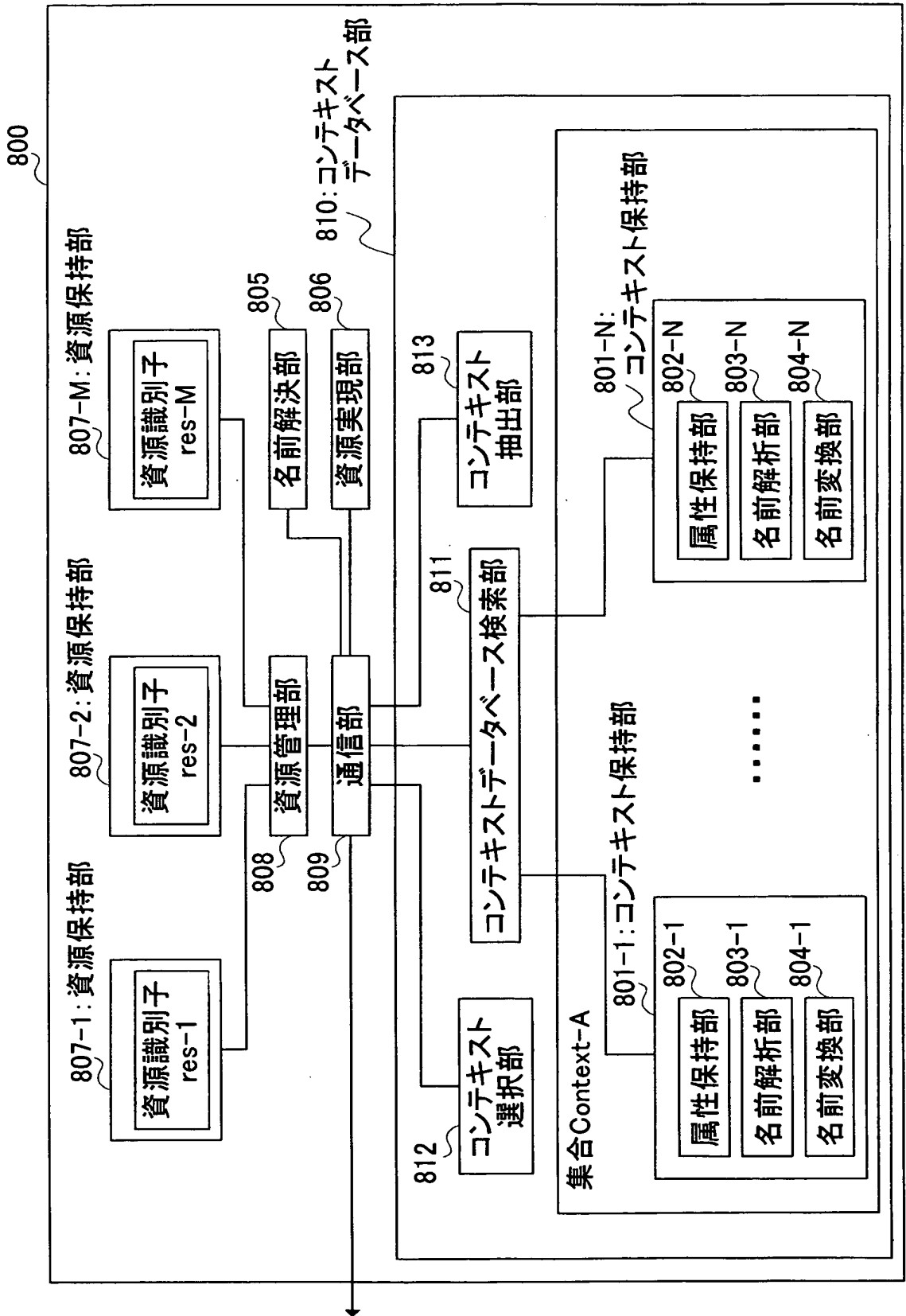
【図 18】



【図 19】



【図 20】



【図 21】

```
Set<Ref<Class Context>>subcontext:
Context->query(subcontext.
    "select dlstinct x form Context-A where \
        x.property.owner="Smith" \
        and x.property.ndata>95.3.16 \
        and x.property.category="picture\"")
):
```

**【書類名】 要約書****【要約】**

**【課題】** データベースからコンポーネント化されたマルチメディアデータ処理部を取り出すとき、マルチメディアデータ処理部を一旦データベースから取り出し、そのマルチメディアデータ処理部と他のマルチメディアデータ処理部との条件判定を行う必要があり、マルチメディアデータ処理部の検索に時間がかかる。

**【解決手段】** データベースへマルチメディアデータ処理部本体だけではなく、マルチメディアデータ処理部の条件判定を行う関数の関数エントリも登録しておき、マルチメディアデータ処理部の検索時には、まず関数エントリのみ取り出し条件判定を行い、目的のものであった場合のみマルチメディアデータ処理部本体を取り出す。これによって、マルチメディアデータ処理部の検索の時間が短縮される。

**【選択図】** 図 1



特願 2 0 0 3 - 3 0 5 2 5 8

出 願 人 履 歷 情 報

識別番号

[ 0 0 0 0 0 5 8 2 1 ]

1. 変更年月日

1 9 9 0 年 8 月 2 8 日

[変更理由]

新規登録

住 所

大阪府門真市大字門真 1 0 0 6 番地

氏 名

松下電器産業株式会社